

中原大學資訊工程系 演算法分析第二次機測

Deadline: 6 / 9 / 2023 (星期五)
(限期末考前測完，逾期不得補繳)

【程式設計說明】

1. 每組限 2~3 人，組員須固定，本學期不得任意變更。原則上以專題組員為主。
2. 組員應合作共同解題，但嚴禁跨組合作。
3. 程式設計必須使用 Python 程式語言，版本請採用目前最新版本(原則上，請直接下載與安裝 Anaconda)。
4. 可參考課本、參考書籍或網站資料等進行解題，解題方法及演算法不限，但絕對嚴禁抄襲他組程式，組員均有責任保護程式不被他組抄襲。若發現抄襲屬實，兩組均以零分計。
5. 輸入與輸出採用標準格式或讀寫檔案方式進行。
6. 每一支程式均須附上組員姓名及學號，例如：

```
# 演算法分析機測  
# 學號: 10927XXX / 10927XXX  
# 姓名: 陳○○ / 林○○  
# 中原大學資訊工程系
```

程式命名依該組學號在前之同學 [學號+題號] 為原則。例如：

```
10927001_1.py  
10927001_2.py
```

【機測須知】

1. 評分以解題成功之題數多寡與執行時間決定。
2. 程式必須能處理不同的輸入資料(但輸入格式與範例相同)，並輸出正確結果(輸出格式必須與範例相同)，組員應能說明程式設計內容，方可視為成功。程式的輸出結果錯誤、輸出格式與範例不符、或在執行後超過 5 秒(以每筆測資為基準)仍未結束，均視為失敗。若程式測試失敗給予基本分數，未繳交程式則以零分計。
3. 本機測於規定之期限前，各組應攜帶程式原始碼至電學大樓 603 室找助教測試(電話：265-4726)，每組限繳交一次，不可分題或多版本繳交，逾期不得補繳。
4. 助教將使用不同之輸入資料作為測試與評分依據，同學應在繳交前充分測試程式。
5. 機測成績納入學期平時成績計算，請同學把握！

指導教授: 張元翔

I. 最長遞增子序列 (Longest Increasing Subsequence)

電腦科學領域中，**最長遞增子序列** (Longest Increasing Subsequence) 問題，簡稱 LIS 問題，是具有代表性的問題。LIS 問題描述如下：給定一個數字序列，目的是找到其中的一個子序列，這個子序列必須從小到大排序，且長度必須最長。舉例說明：

給定數字序列： $\langle 10, 22, 9, 33, 21, 50, 41, 60, 80 \rangle$

則 LIS 為： $\langle 10, 22, 33, 50, 60, 80 \rangle$

試採用**動態規劃法** (Dynamic Programming) 的設計策略進程式設計，解決 LIS 問題。

輸入說明：

輸入含有多組測試資料，每組測試資料有 2 列。每組測試資料的第一列有 1 個整數 n ($n \leq 1,000$)，0 代表結束。接下來的第二列為數字序列，數字均為正整數，中間用空格隔開。

輸出說明：

對每一組測試資料，輸出 LIS 的最佳解，包含 LIS 的長度與 LIS。LIS 中間用逗號隔開，最後一個數字不能有逗號。

輸入範例：

```
9
10 22 9 33 21 50 41 60 80
8
10 9 2 5 3 7 101 18
0
```

輸出範例：

```
Case 1
Length of LIS = 6
LIS = 10, 22, 33, 50, 60, 80
Case 2
Length of LIS = 4
LIS = 2, 3, 7, 101
```

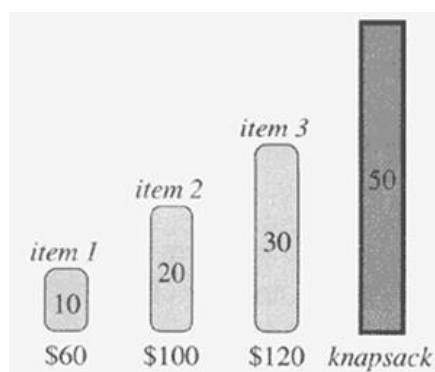
II. 0-1 背包問題 (0-1 Knapsack)

電腦科學領域中，**0-1 Knapsack 問題** (或稱為 Bin-Packing 問題) 是具有代表性的問題。問題描述如下: 有一個小偷到一家店內偷東西，他發現店內有 n 項物品，每項物品各有不同價值及不同重量。小偷的目的是帶走總價值最高的物品，但他能帶走的**背包** (Knapsack) 有重量的限制。

試設計程式解決 0-1 背包問題 (即每項物品僅能**取走或不取**，無法取走部分)，並須求得最佳解 (Optimal Solution)。

輸入說明:

輸入物品 Knapsack 重量 W 及物品總數 n (0 0 代表結束)，接著分別是各項物品的重量及價值 (均為正整數)。以下為輸入範例:



輸出說明:

求出可能之最高總價值，並列出取走物件的編號 (須按編號由小到大順序排列，並以逗號隔開，最後一個物件不能有逗號)。

輸入範例:

```
50
3
10 60
20 100
30 120
0 0
```

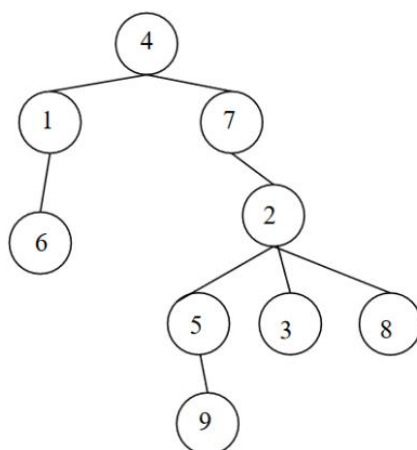
輸出範例:

```
Total Value = 220
Take Items 2, 3
```

III. 有根樹 (Rooted Tree)

本題有關有根樹 (Rooted Trees)。在一棵 n 個節點的有根樹中，每個節點是以 $1 \sim n$ 的不同數字來編號，描述一棵有根樹必須定義節點與節點之間的親子關係。一棵有根樹只有一個節點沒有父節點 (Parent)，此節點被稱為根節點 (Root)。除了根節點以外的每一個節點都恰有一個父節點，而每個節點被稱為其父節點的子節點 (Child)。有些節點沒有子節點。這些節點稱為葉節點 (Leaf)。當有根樹只有一個節點時，這個節點是根節點，同時也是葉節點。

以下圖為例說明，我們將父節點畫在子節點之上，中間畫一條邊 (Edge) 連結。本圖包含一棵 9 個節點的有根樹；其中，節點 1 為節點 6 的父節點，而節點 6 為節點 1 的子節點；節點 5、3、8 為節點 2 的子節點。節點 4 沒有父節點，所以節點 4 是根節點；而節點 6、9、3 與 8 都是葉節點。



有根樹中的兩個節點 u 和 v 之間的距離 $d(u, v)$ 定義為兩節點之間邊的數量。如圖所示， $d(7, 5) = 2$ 、 $d(1, 2) = 3$ 。對於有根樹中的節點 v ，我們以 $h(v)$ 代表節點 v 的高度，可以定義為節點 v 與下方最遠的葉節點之間的距離，而葉節點的高度定義為 0。因此，節點 6 的高度為 0，節點 2 的高度為 2，節點 4 的高度為 4，依此類推。此外，我們定義 $H(T)$ 為有根樹 T 中所有節點的高度總和。

給定一個有根樹 T ，請設計程式找出 T 的根節點與高度總和。

輸入說明：

第一行為一個正整數 n ($n \leq 1,000$)，代表有根樹的節點個數，節點的編號為 $1 \sim n$ (0 則代表結束)。接著有 n 行，第 i 行的第一個數字 k 代表節點 i 有 k 個子節點；第 i 行接下來的 k 個數字就是這些子節點的編號，中間以空白隔開。

輸出說明：

就每組輸入列出結果，包含：根節點的編號與高度總和。

輸入範例:

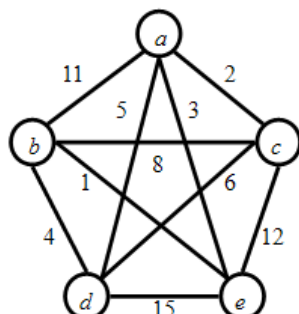
9
1 6
3 5 3 8
0
2 1 7
1 9
0
1 2
0
0
7
0
2 6 7
2 1 4
0
2 3 2
0
0
0

輸出範例:

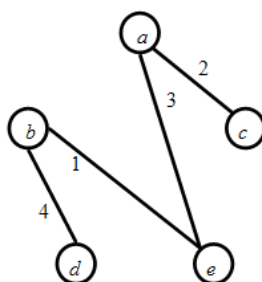
Rooted Tree 1
4
11
Rooted Tree 2
5
4

IV. 最小生成樹 (Minimum Spanning Trees)

給定一無向圖 $G = (V, E)$ ，範例如下：



所謂「生成樹」(Spanning Tree) 是指在圖形中找到的子圖 (Subgraph) 必須符合兩個條件：
(1) 該子圖必須符合樹的定義；(2) 該子圖必須涵蓋所有頂點。最小生成樹 (Minimum Spanning Tree) 則是指在圖形中找出一個具有最小成本 (Minimum Cost) 的 Spanning tree。以上圖為例，則找到之最小生成樹如下：



其最小成本 Minimum Cost = 1 + 2 + 3 + 4 = 10。若以暴力法方式決定其最小生成樹 (Minimum Spanning Tree)，首先試決定所有可能之生成樹個數。此外，試寫程式決定其最小生成樹，並計算其最小成本。

輸入說明：

每組輸入含：

Vertices 個數 n (原則上頂點數不超過 20)

Weight Matrix (其中，每一個 weight 均為正整數，並以空格隔開，若無 Edge 則以 “-” 代表。)

Vertices 個數為 0 時代表結束。

輸出說明：

輸出所有可能之生成樹個數及最小生成樹之成本。

輸入範例 (見上圖)：

5

0 11 2 5 3
11 0 8 4 1
2 8 0 6 12
5 4 6 0 15
3 1 12 15 0
0

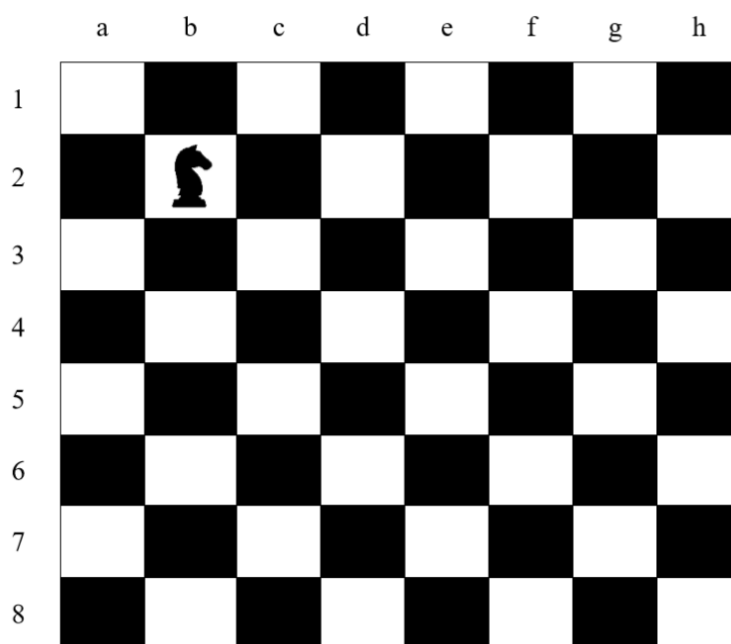
輸出範例：

Number of Spanning Trees = 125

Minimum Cost = 10

V. 西洋棋騎士 (Chess Knight)

西洋棋 (Chess) 是一種二人對弈的戰術棋盤遊戲，也是世界上最流行的遊戲之一。在此，讓我們探討西洋棋騎士 (Knight) 的移動問題，如下圖。西洋棋盤是一個 8×8 的棋盤，每一列使用 1~8 編號；每一行則使用 a~h 編號。



我們想要解決的問題是：「給定兩個位置 X 與 Y，若騎士從 X 到 Y 至少需要走幾步？」

舉例說明，若想將騎士從 b2 移到 c3，至少需要 2 步，即先將騎士從 b2 移到 d1，再從 d1 移到 c3。另一種走法，是先將騎士從 b2 移到 a4，再從 a4 移到 c3，但移動的步數相同。

輸入說明：

兩個西洋棋的座標位置。每個座標位置是由一個小寫英文字母 (a~h) 與一個數字 (1~8) 組成；00 代表結束。

輸出說明：

騎士至少需移動的次數。

輸入範例：

b2 c3

a1 b2

a1 h8

0 0

輸出範例:

From b2 to c3, Knight Moves = 2

From a1 to b2, Knight Moves = 4

From a1 to h8, Knight Moves = 6